ESP: Echo-Skeleton Perception

Stateful, Event-Driven Screen Sensing for Blind Language Models

Olajide Al-ameen
Bad Theory Labs
hello@badtheorylabs.com

July 2026 — Thesis Draft v1

Abstract

Every serious computer-use agent today perceives the screen the same way: capture a full screenshot, hand it to a large vision–language model, act, sleep, capture again. The loop is expensive, stateless, and — critically — blind in the interval between captures. We propose **Echo-Skeleton Perception (ESP)**, a perception layer that lets a small, local, *text-only* language model operate a graphical interface without an image anywhere in the loop. ESP perceives in three passes. A **skeleton** pass decomposes the screen into an adaptive region tree and actively probes it — hit-testing, hover response, cursor change — to discover structure and affordances, the way echolocation discovers a cave. A **populate** pass runs classical OCR only inside discovered regions, so the skeleton wakes up labeled with exactly what the page says, where it says it. An **echo** pass then streams millisecond-scale change events against the persistent skeleton — a progress bar advancing, a modal appearing, a click that silently failed — so the agent *senses* the interface evolving instead of repeatedly re-photographing it. The reasoner holds a labeled skeleton as world state, receives ten-token deltas, and acts through schema-strict tool calls: perception becomes mechanical and nearly free, and reasoning becomes the only place tokens are spent. This inverts the field's direction of travel, which is toward larger vision models consuming denser screenshots more often. This is a thesis paper in the strict sense: we contribute the architecture, four falsifiable hypotheses with numbers attached, and a pre-registered experimental plan whose decision rule is fixed here, before the first line of the prototype exists. We report no results and make no empirical claims.

1 Introduction

A person using a computer does not look at the screen a thousand times. They look at it roughly once. They form a model of the page — here is the form, there is the button, that corner has an ad — and from then on they *notice* rather than *observe*: the spinner is still going; the upload bar just finished; my click did nothing, click again. Perception after the first glance is almost entirely the perception of change.

Today's computer-use agents are built the other way around. Each step of an agentic episode begins with a fresh screenshot of a screen that is, almost always, nearly identical to the previous one. The screenshot is consumed by a large vision–language model at a cost of one to two thousand tokens. The model re-derives the structure of the page from pixels, re-locates the elements it already found, acts once, sleeps for a guessed interval, and photographs the world again. State lives nowhere except, precariously, in the context window; between captures the agent is completely blind.

Three structural pathologies follow directly from this design, and no amount of model scale fixes them, because they are properties of the loop rather than of the network:

1. **Cost scales with steps, not with change.** A twenty-step task pays for twenty perceptions of largely the same screen. The information genuinely new at each step — one region changed, one dialog appeared — is a rounding error inside the tokens billed.

2. **Statelessness.** The page structure is re-inferred from scratch at every step, and re-inferred slightly differently each time. Agents lose track of elements they have already used; identical buttons swap identities between frames.

3. **Blindness between actions.** An agent that clicks a button learns whether the click worked one full capture later, if it learns at all. Silent failures — the click that misses by four pixels, the button that was not yet enabled — are the leading cause of derailed episodes, and the standard mitigation is a guessed `sleep()`.

Our starting observation is that the screen was never a natural object of pixel-first perception. A rendered interface is not a photograph of the world; it is the output of a layout engine. It is mostly flat rectangles of nothing, structured by a small number of meaningful regions, labeled by a small amount of text, and animated by sparse, spatially localized change. Every one of those four properties has a classical, decades-old, essentially free sensor:

Property of screens	Classical sensor
mostly empty, locally detailed	adaptive spatial decomposition (quadtrees [1])
affordances announce themselves	hit-testing, hover response, cursor change
meaning is carried by text	OCR inside known rectangles [5]
change is sparse and localized	damage tracking; VNC/RDP dirty rectangles [4]

None of these requires a neural network at perception time. The last one is not even new computation: remote-desktop protocols have transmitted exactly this change stream, as their wire format, since the 1990s. The pieces have been lying in plain sight; what has been missing is the decision to wire them together into a sense organ and hand it to a model that cannot see.

Echo-Skeleton Perception (ESP) is that organ. It perceives in three passes — *skeleton*, *populate*, *echo* — described in Section 3, and it presents the interface to a small text-only language model as a persistent, labeled, structured world plus a stream of change events. The model reasons over structure it can trust and spends tokens only on decisions. The name is a deliberate pun: ESP gives a model with no eyes something like extrasensory perception — a cave sensed through echoes, flown through in the dark.

1.1 Thesis

> *A small text-only language model, given a persistent labeled skeleton of the screen and a millisecond-scale stream of change events, can complete realistic interactive tasks currently assumed to require vision–language models consuming full screenshots — at a fraction of the perception cost, with faster and more reliable action verification.*

1.2 Contributions

This paper is written before its own prototype, deliberately, so that the claims are fixed in place before results exist to tempt us into adjusting them. It contributes:

1. **An architecture** (Section 3): the three-pass perception stack and the agent contract that makes a screen legible to a blind reasoner.

2. **A position** (Section 2): a precise accounting of which ingredients are old — most of them — and what exactly is new: the assembly of persistent skeleton, active probing, and delta stream as the *sole* perception of a text-only agent.

3. **Four falsifiable hypotheses with numbers attached** (Section 5): sufficiency, economy, verification, and robustness.

4. **A pre-registered experimental plan and decision rule** (Section 6), including what we ship if the thesis partially fails and the commitment to publish if it fully fails.

We report no experimental results. Every empirical sentence in this paper is a prediction.

2 Related Work, and What Is Actually New

Nearly every ingredient of ESP exists somewhere in the literature or in decades-old systems software. We think the assembly does not. This section is written to survive a hostile reviewer: we name what we did not invent first, and defend the smaller, sharper claim that remains.

Element marking. Set-of-Mark prompting [6] overlays numbered boxes on detected UI elements and reliably improves grounding for vision agents; strong OSWorld scaffolds report gains of several points from marking alone [10]. ESP's numbered skeleton nodes inherit this referencing idea. The difference is that our marks are derived without any vision model and consumed without any image: the numbers do not annotate a picture, they *replace* it.

Screen parsing. OmniParser [7] and its relatives convert screenshots into structured element lists for a downstream LLM, using small trained detectors and captioners. ESP shares the stance that perception belongs outside the reasoning model, and pushes it one step further — classical decomposition and *active probing* in place of learned perception — and one dimension further: parsers describe a frame; ESP maintains a world. OmniParser has no temporal axis. Pass 3 is precisely the axis it lacks.

DOM and accessibility-tree agents. Text-only agents acting on DOM or accessibility trees — the WebArena lineage [8, 9] — are the existence proof that blind models can operate structured pages at all, and they are the natural upper baseline for our experiments. Their known failure mode is *trusting the tree*: canvas-rendered applications, Flutter web, games, and custom widget toolkits expose trees that are empty or lying. ESP probes the rendered surface itself. Hit-testing and repaint-watching do not care how the pixels were produced, so ESP degrades gracefully exactly where tree agents die — the content of hypothesis H4. Where a trustworthy tree exists, ESP can consume it as one more cheap sensor; the architecture is agnostic.

End-to-end vision agents. UI-TARS [11] and its successors unify perception and action inside one large vision–language model and, with scaffold support, lead OSWorld-class benchmarks [10]. We are not competing with these systems on their turf and this paper claims no such thing. ESP is the opposite bet, aimed at the regime end-to-end vision prices out — small local models on consumer hardware — and at a question the benchmark race does not ask: *how much of the gap between small and large agents is perception, rather than reasoning?* If a 4B model with mechanical senses closes most of the distance to a far larger vision model on non-visual tasks, that is a finding about the field's cost structure, not just about our system.

Damage tracking. Dirty-rectangle computation is among the oldest ideas in systems graphics: compositors track damage regions, and the VNC Remote Framebuffer protocol [4] transmits *only changed rectangles* as its wire format. We claim zero novelty here. ESP's observation is sociological as much as technical: an agent can subscribe to this pre-existing stream as a sense organ, and apparently no one has.

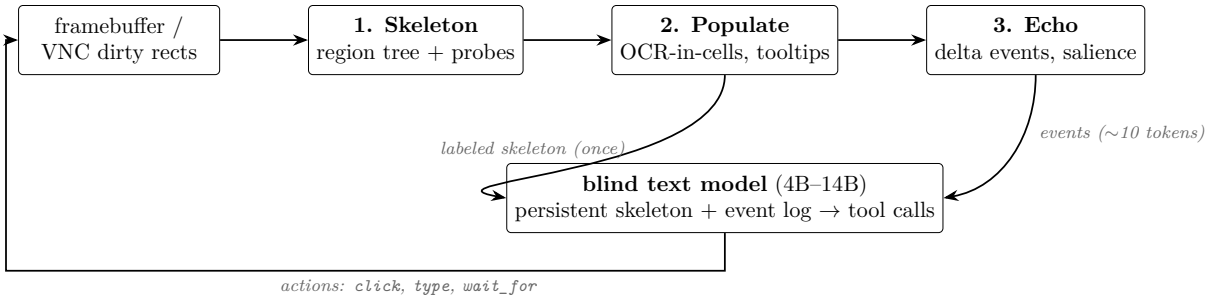

Figure 1: ESP in one picture. Classical code touches pixels; the model never does. The skeleton is transmitted once; afterward the model perceives only change. Actions feed back into the surface, and their echoes — or telling silence — return as events.

Active perception. Robotics has distinguished passive sensing from active exploration since the 1980s [2, 3]: an agent that controls its own sensors gathers information passively unavailable. ESP's probe loop — ping the surface, read the echo, refine where the echoes are interesting — is active perception applied to a screen. We have found no prior agent system that treats hover response, cursor change, and hit-testing as a deliberate sensing modality rather than as incidental automation plumbing.

The residual claim. What remains after these deductions is exactly this: **a persistent labeled skeleton, built by classical decomposition and active probing, maintained by a dirty-region event stream, serving as the sole perception of a small text-only reasoner.** No shipped system or paper we are aware of implements this combination or evaluates whether it suffices. Meanwhile the field's momentum points the other way — larger vision models, denser screenshots, higher capture rates — which makes the cheap direction not merely unexplored but structurally neglected. Theses like this one are how small laboratories are supposed to earn their keep.

3 The ESP Architecture

We describe the three passes, then the contract they jointly present to the model. Throughout, the design constraint is severe on purpose: *no neural network anywhere in perception.* Wherever we are tempted to solve a perception problem with a model, we treat that temptation as evidence the mechanical design is not finished.

3.1 Pass 1: Skeleton — structure from echoes

A single framebuffer grab is decomposed by recursive subdivision in the manner of a quadtree [1]: a region of uniform appearance survives as one large cell; a region containing detail splits, and splits again, until cells hug content boundaries. Cell size itself becomes information — where cells are small and dense, something is there; where one lazy cell covers a quarter of the screen, nothing is. A 1080p page of two million pixels typically reduces to a few hundred cells, each describable in one line of text. This is the "field that deforms around content": the skeleton is not a picture of the page but the shape the page presses into a grid.

The skeleton is then *probed*, and this is where perception becomes active [2]. The probe suite consists of questions the windowing system and browser answer for free, without any image:

- **Hit-tests**: what object occupies coordinate (x, y)?
- **Hover echoes**: sweep the pointer; watch for cursor change (arrow → hand is the system announcing *clickable*), tooltip emission, and hover-triggered repaints.

- **Focus walks**: keyboard traversal order, which discovers editable fields and their sequence.

Probe results attach affordance flags to cells: *clickable*, *editable*, *scrollable*, *inert*. Probing is rate-limited and prioritized by the tree itself — dense structure is probed first, blank space never — and the map is cached: a control probed once is known for the rest of the session. Formally, the pass yields a tree $S = (N, \sqsubset)$ of nodes $n = (geom, afford, label)$ with *label* still empty; containment $\sqsubset$ gives the model "the button inside the form inside the left panel" for free.

3.2 Pass 2: Populate — the skeleton learns to read

Classical OCR [5] runs *inside discovered cells only*. This inversion matters twice. It is dramatically cheaper than whole-screen OCR, because text-bearing rectangles are already isolated; and it is more accurate, because each recognition runs on a cropped, single-purpose region rather than a busy page. Most importantly, every recovered string is *born attached to a node*: the skeleton does not receive a bag of words but learns what each of its bones says. The distinction between two identical rectangles — `[Submit]` and `[Cancel]` — is not geometric information and cannot be sensed by any amount of proximity; it must be read. Pass 2 is where the irreducible symbolic content of the page, and only that content, enters the system.

Icons — shapes that say nothing — are resolved by the probe loop instead: a hover dwell elicits the tooltip, the tooltip names the icon, and the name is cached in the skeleton permanently. One slow probe converts an unlabeled glyph into a labeled control for the rest of the session.

After Pass 2 the agent possesses, as a few hundred lines of structured text, a claim most screenshot agents cannot make: it knows *exactly* what the page says, where each statement lives, and what each region will do when touched.

3.3 Pass 3: Echo — perception of change

The screen is diffed continuously at millisecond scale: directly from the VNC/RDP dirty-rectangle stream [4] where one is available — the protocol computes our sense for us as a by-product of existing in the world — or otherwise by downscaled frame differencing on CPU, which is arithmetic, not inference. Changed rectangles are mapped onto skeleton nodes and emitted as typed events:

```
changed(node 12, text: "45%" -> "52%")      # progress advancing
appeared(node 30, modal, children=[31..36]) # dialog opened; region re-
   skeletoned
vanished(node 30)                            # dialog closed
quiet(node 12, 800ms)                        # loading region has settled
no_change(after=click(node 14), window=600ms) # the click did nothing
```

Two disciplines keep the stream sane. **Salience filtering**: a region that changes constantly — video, animated advertisement, spinner — is classified as animation and muted after a single notification (`animated(node 8, muted)`). Real echolocation solves the same problem; the bat must not deafen itself with its own wings. **Action windows**: immediately after the agent acts, the echo layer opens an expectation window over the plausible consequence region. Change inside the window is reported as confirmation; silence is reported as `no_change` — an explicit, first-class percept. This single event type addresses the leading cause of derailed agent episodes: the silently failed action, currently discovered a full screenshot later or never.

Pass 3 also converts *waiting* from a guessed timer into a perception primitive. The agent does not sleep 3000ms and hope; it subscribes to `quiet(node)` and is woken by the world.

3.4 The agent contract

The model — deliberately small, local, text-only — receives:

1. the labeled skeleton, once, at session start (and incremental sub-skeletons when new surface appears);
2. the event stream, thereafter, at a few dozen tokens per event;

and emits schema-strict tool calls against node identity, never against pixels:

```
click(node)        type(node, text)      scroll(node, dy)
wait_for(node, quiet | changed | appeared, timeout)
probe(region)      read(node)            focus_next()
```

The contract is deliberately identical in shape to function calling. This has a consequence beyond elegance: **ESP episodes are tool-use training data.** A model post-trained for schema-strict tool calling is pre-adapted to ESP with no vision training of any kind, and every ESP session — skeleton, events, calls, outcomes — is a supervised trace in exactly the format our existing tool-use corpus uses. Perception and training strategy close into one loop.

4 A Worked Session

To make the contract concrete, here is one hypothetical episode — upload a file and confirm success — as the model would experience it. It is an illustration of the intended interface, not a result. Perception layer output is shown plain; the model's emitted tool calls are marked `>>`.

```
SKELETON (session start, 214 nodes, abbreviated):
  node 3   region  "Upload your document"      inert
  node 7   button  "Choose file"               clickable
  node 9   text    "No file selected"          inert
  node 14  button  "Upload"                    clickable, disabled
  node 22  region  <empty, 60% of viewport>    inert

>> click(node 7)
  appeared(node 40, dialog "Open", os-native)
>> (native file dialog handled by host; path returned)
  changed(node 9, text: "No file selected" -> "report.pdf")
  changed(node 14, afford: disabled -> clickable)     # Upload became
     live

>> click(node 14)
  changed(node 14, text: "Upload" -> "Uploading...")
  appeared(node 51, progressbar, region under node 14)
  changed(node 51, text: "0%" -> "18%")
  changed(node 51, text: "18%" -> "47%")

>> wait_for(node 51, quiet, timeout=30s)
  changed(node 51, text: "47%" -> "100%")
  quiet(node 51, 900ms)
  vanished(node 51)
  appeared(node 60, banner "Upload complete", green region)

>> read(node 60)
  "Upload complete. Document ID 8842."
>> done(success, detail="Document ID 8842")
```

Three properties of this trace are worth naming, because each is a claim the experiments must test rather than a fact we assert:

- **The model never saw an image**, yet it distinguished a disabled button from a live one, read a filename it did not type, tracked a progress bar to completion, and confirmed success from banner text. Every one of those percepts arrived as a labeled event against a node it already held.
- **Waiting was sensed, not guessed.** `wait_for(quiet)` returned when the world settled, not when a timer expired. An agent that guesses `sleep(5s)` either wastes four seconds or wakes at 47% and misfires.
- **The whole episode's perception cost** is one skeleton plus roughly a dozen event lines. A screenshot agent would have paid for perhaps eight full captures — most of them re-encoding the same static upload form — to see the same events. This is the arithmetic behind hypothesis H2.

The counterfactual sharpens H3. Suppose the `click(node 14)` had missed — a mispredicted coordinate, a not-yet-enabled control. A screenshot agent proceeds to `wait_for` and then photographs a screen with no progress bar, and must *infer* from absence that something went wrong. The ESP agent receives, within the action window, an explicit `no_change(after=click(node 14))` and retries immediately. The failure is a percept, not a deduction from a later photograph.

5 Hypotheses

Stated before implementation, each falsifiable, each with a number, none supported by evidence we currently possess. They are ordered so that the cheapest-to-test and most load-bearing come first.

H1 — Sufficiency. A text-only model in the 4B–14B range, driving ESP, completes realistic multi-step web tasks (form filling, waiting on asynchronous operations, confirming outcomes) at a success rate within **10 points** of the same base model driving a full accessibility-tree interface, and *above* a screenshot baseline that uses an open vision model of comparable parameter count. Rationale: if structure-plus-text-plus-change is a sufficient statistic for non-visual tasks, the blind agent should nearly match the tree agent (which sees perfect structure) and beat the small vision agent (which sees pixels but reasons no better). *Falsified if* ESP trails the tree agent by more than 10 points or fails to beat the vision baseline.

H2 — Economy. Perception cost per completed task, measured in tokens delivered to the model, is at least 5× **lower** under ESP than under a per-step screenshot baseline. Rationale: a screenshot re-transmits the whole screen every step; ESP transmits it once and streams only deltas, and deltas are where the new information actually is. *Falsified if* the median token ratio across tasks is below 5×, for instance because skeletons must be re-sent too often or event streams are too chatty after salience filtering.

H3 — Verification. On tasks deliberately seeded with unreliable controls — buttons that fail silently a known fraction of the time — the echo layer's action windows raise task success by at least **15 points** over an otherwise identical ESP agent with Pass 3 disabled. Rationale: instant `no_change` feedback converts a silent failure from an undetected derailment into an immediate retry. This is the one hypothesis whose mechanism is useful even if H1 fails, which is why it is tested independently. *Falsified if* the Pass-3 ablation matches full ESP on flaky-control tasks within 15 points.

H4 — Robustness. On canvas-rendered pages whose accessibility tree is empty or misleading, ESP retains at least **70%** of its structured-page success rate, while a DOM-tree agent on the same pages degrades toward zero. Rationale: probing the rendered surface is independent of how the surface was produced, so the sensor that reads a native form also reads a canvas one; the tree agent has nothing to read. *Falsified if* ESP loses more than 30% of its success on canvas pages, indicating its perception secretly depends on tree-like structure after all.

H1 and H2 decide whether ESP is a system. H3 decides whether at least its echo layer is a product regardless. H4 decides whether it beats the obvious cheaper alternative (just read the tree) where that alternative is supposed to be strongest — and where, in the real world, it most often is not.

6 Pre-Registered Experimental Plan

The plan is fixed here, before the prototype, so that no result can retroactively edit the question it answered. Deviations, if any prove necessary, will be reported as deviations.

Environment. Playwright-driven Chromium. Playwright is chosen because it simultaneously provides the ground-truth action channel, the probe APIs (hit-testing, hover, focus order), and — for the baselines — the accessibility tree and full-page screenshots. It can also log every DOM mutation with timestamps, which gives us exact ground truth for scoring echo precision and recall (Section 6). All pages are frozen to fixed snapshots for reproducibility.

Tasks. Three tiers of increasing adversarial pressure.

- **T1 — deterministic local pages.** Form fill; multi-step wizard; file upload with a progress bar; delayed modal confirmation. Fully controlled, used for H1 and H2 and for debugging the perception layer.
- **T2 — frozen real pages.** WebArena-lite-style [8] tasks on permissively licensed real sites, snapshotted. Tests whether ESP survives contact with real-world layout mess.
- **T3 — adversarial pages.** Canvas-rendered UIs with empty trees (H4); pages with animated distractors (salience stress); pages with silently failing controls (H3).

Agents and conditions. Two reasoners — `qwen3:4b` and a 9B-class model (Ornith-1.0-9B [12] or a Qwen3 sibling) — each under four perception conditions:

(a) ESP full (skeleton + populate + echo);

(b) ESP with Pass 3 disabled — the H3 ablation;

(c) accessibility tree — the DOM-agent *upper* baseline, sees perfect structure;

(d) screenshots + a comparably sized open vision model — the paradigm baseline.

Holding the reasoner fixed across conditions isolates perception as the only variable, which is the entire point: any success gap between (a) and (c) is attributable to ESP's perception, not to the model's intelligence.

Metrics.

- **Task success** (primary): task-specific goal predicate, checked from ground-truth page state, not from the agent's self-report.
- **Perception tokens per completed task** (for H2).
- **Wall-clock per task**, including probe latency, so the cost of active perception is charged honestly.
- **Action-failure recovery rate** (for H3): fraction of silent failures the agent detects and retries.
- **Echo precision / recall**: events surfaced vs. events relevant, scored against Playwright's exact mutation log, so we can tell a quiet stream (good) from a blind one (fatal).

Decision rule. Fixed in advance:

- **H1 and H2 supported** → build the full system and fold ESP-native traces into the flagship tool-use model; ESP becomes a lab product line.
- **H1 fails but H3 supported** → ship the echo layer alone as an action-verification add-on for conventional screenshot agents. Instant silent-failure detection is independently valuable and vendor-agnostic.
- **H4 supported in isolation** → position ESP specifically for canvas/game/custom-widget automation, the tree agents' graveyard, even if general sufficiency is marginal.
- **All hypotheses fail** → publish the negative result with the full trace corpus. The failure profile of blind agents on realistic pages is itself unmeasured, and a rigorous negative is worth more than a buried one.

There is no branch of this rule under which the effort produces nothing publishable. That is by design.

7 Failure Modes We Expect

A thesis that cannot say where it dies is not falsifiable. These are the six places we expect ESP to be attacked, with our honest read on each.

1. **Skeleton fuzz.** Recursive decomposition segments by *appearance*, and cell boundaries are not element boundaries. A gradient hero image over-segments; a flat card holding three controls under-segments. Probing partly rescues this — affordances cluster where controls are regardless of cell edges — but a skeleton that carves a button in half will confuse a blind model that a screenshot would not. Mitigation is snapping cell boundaries to strong edges before probing; whether that is enough is empirical.
2. **Lying echoes.** The cursor-change convention is a convention, not a contract. Some clickable elements never change the cursor (bare `onClick` on a `div`); some inert elements do. Affordance flags will therefore carry false positives and false negatives, and a blind model cannot cross-check them against appearance. We expect to need a confidence field on affordances and a reasoner tolerant of occasional wrong flags.

3. **OCR noise.** Stylized fonts, low contrast, overlapping text, and non-Latin scripts will misread, and a blind model cannot eyeball-correct a garbled label the way a person glancing at the button would. This is the sharpest limit on Pass 2 and the strongest argument for keeping a vision fallback available for text specifically, even in an otherwise imageless system.

4. **Probe latency.** Hover sweeps are serial and can cost hundreds of milliseconds each; a probe-hostile page with hundreds of ambiguous cells could take seconds to skeleton. The cost amortizes across a session (probe once, know forever), but first-glance latency on a fresh complex page is a real tax that H2's wall-clock metric will expose rather than hide.

5. **Event floods.** Salience filtering is a heuristic, and heuristics have adversaries. A page that animates genuine content — a live-updating dashboard, a stock ticker that is also the task target — forces a choice between muting information the agent needs and drowning the agent in it. We do not have a principled solution, only a tunable threshold, and we expect T3 to punish it.

6. **The hard ceiling.** Some tasks are irreducibly visual: interpreting a chart, judging whether two colors match, reading text baked into an image, solving a CAPTCHA designed to defeat exactly this pipeline. ESP does not address these and we will not pretend otherwise. The honest question is not whether a visual ceiling exists — it does — but how large the non-visual majority of real computer work is beneath it. Our experiments measure that majority incidentally, and if it turns out to be small, that is a finding against the whole thesis and we will report it as one.

The through-line: ESP trades the generality of vision for the cheapness of mechanism, and every failure mode above is a facet of that one trade. The thesis is the wager that, for the work people actually automate, the trade is worth it.

8 Relation to the Laboratory's Program

Bad Theory Labs builds small open-weight models that run on consumer hardware. ESP is that thesis applied to perception rather than to reasoning: it removes the single most expensive organ — the vision–language model — from the agent loop entirely, and leaves behind exactly the kind of model the laboratory already trains, a small, local, schema-strict, tool-calling text model.

The fit is not thematic but mechanical, along three seams:

- **The agent contract is function calling** (Section 3.4). ESP needs no new model capability; it needs the tool-use capability already central to the flagship program. A model good at BFCL-style calling is, by construction, most of the way to driving ESP.

- **ESP sessions are training data.** Each episode is a supervised trace in the exact schema of the tool-use corpus. Perception generates its own future training signal, and the browser-verification infrastructure built for frontend work doubles as the ESP environment.

- **The hardware story is the product story.** If the hypotheses hold, a laptop runs the entire agent — classical skeleton and echoes in a few thousand lines of ordinary code, a small model reasoning over them — with no GPU-class vision model anywhere in the loop. That is the consumer-hardware thesis made literal: not a small model pretending to be a big one, but a small model that never needed the big one's eyes.

ESP is therefore not a side quest. It is the perception half of the same bet the flagship model is the reasoning half of, and the two halves are trained by the same data in the same format on the same infrastructure.

9 Conclusion

The field is racing toward bigger eyes: larger vision–language models consuming denser screenshots, captured more often, at higher cost. We propose the opposite organ. A rendered screen is not a photograph of the world — it is structured, sparse, textual, and locally animated, the deliberate output of a layout engine — and there is a decades-old, essentially free mechanical sensor for each of those four properties. Wire them into a persistent skeleton maintained by an event stream, hand that to a small model that cannot see, and the question changes shape. It stops being *"can a small model see the screen?"* — it never sees it — and becomes *"can a small model think about a screen it senses through echoes?"*

That question is falsifiable for the price of a weekend of engineering and zero GPU-hours, which is precisely why we wrote the paper before the prototype. The claims, the numbers, and the decision rule are fixed in place now, so that whatever the experiments say — sufficiency, partial success limited to the echo layer, robustness on the canvas pages where trees fail, or honest defeat — we cannot quietly move the goalposts to meet the result. A bat does not decide where the cave wall is after it hears the echo. Neither will we.

References

[1] R. A. Finkel and J. L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

[2] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[3] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, 1988.

[4] T. Richardson and J. Levine. The Remote Framebuffer Protocol. RFC 6143, IETF, 2011.

[5] R. Smith. An overview of the Tesseract OCR engine. In *Proc. ICDAR*, 2007.

[6] J. Yang et al. Set-of-Mark prompting unleashes extraordinary visual grounding in GPT-4V. arXiv:2310.11441, 2023.

[7] Y. Lu et al. OmniParser for pure vision based GUI agent. arXiv:2408.00203, 2024.

[8] S. Zhou et al. WebArena: A realistic web environment for building autonomous agents. arXiv:2307.13854, 2023.

[9] X. Deng et al. Mind2Web: Towards a generalist agent for the web. arXiv:2306.06070, 2023.

[10] T. Xie et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. arXiv:2404.07972, 2024.

[11] Y. Qin et al. UI-TARS: Pioneering automated GUI interaction with native agents. arXiv:2501.12326, 2025.

[12] DeepReinforce AI. Ornith-1.0: Self-scaffolding open models for agentic coding. Technical report, June 2026. https://deep-reinforce.com/ornith_1_0.html.